

Lec: 5

Multiple Features

Linear regression with multiple Variables

Size (Feet ²)	no. of bedrooms	no. of Floors	Age of home (Year)	Price (\$1000)
X_1	X_2	X_3	X_4	Y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
\vdots	\vdots	\vdots	\vdots	\vdots

n : no. of Features.

$x^{(i)}$: input (Features.) of i^{th} training example.

$x_j^{(i)}$: Value of Feature j in i^{th} training example.

ex: From the table $n = 4$

$$x^{(i)} = [X_1 \text{ } X_2 \text{ } X_3 \text{ } X_4]$$

$$x_j^{(i)} = 2104 \text{ } 3 \text{ } 1 \text{ } 45$$

Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$

Parameters: $\theta_0, \theta_1, \dots, \theta_n$

Cost Function: $J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

Gradient descent:

($n \geq 1$)

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \dots, \theta_n)$$

}

(Simultaneously update for every $j=0, \dots, n$)

$$\rightarrow \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_0(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\rightarrow \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_0(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\rightarrow \theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_0(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

...

Feature Scaling

Idea: Make sure Features are on a similar scale.

- 1 Scale by dividing the actual Feature Value by the Feature maximum value.

$$\text{ex: } x_1 = \text{Size (0-2000 Feet}^2) \rightarrow x_1 = \frac{\text{Size (Feet}^2)}{2000}$$

$$x_2 = \text{number of bedrooms (1-5)} \rightarrow x_2 = \frac{\text{no. of bedrooms}}{5}$$

get every Feature into approximately

$$0 \leq x_i \leq 1$$

range.

2] Scale by replacing x_i by $\frac{x_i - \mu_i}{S}$ where $S = \frac{\text{Max} - \text{Min}}{\text{Average}}$

S : Standard deviation.

μ_i : average mean.

→ replace x_i with $x_i - \mu_i$ to make Features have approximately Zero mean (Do not apply to $x_0 = 1$).

ex: $x_1 = \frac{\text{Size} - 1000}{2000}$

→ $-0.5 \leq x_1 \leq 0.5$

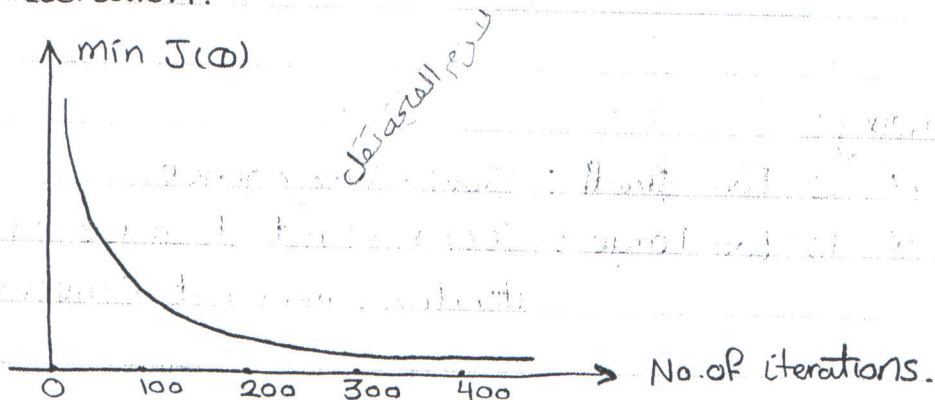
$x_2 = \frac{\text{*bedroom} - 2}{5}$

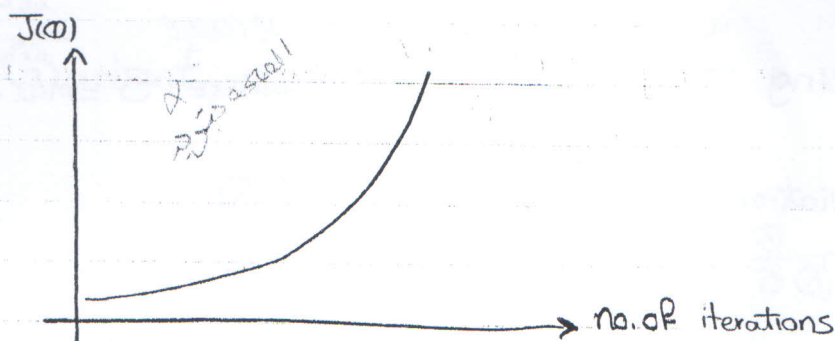
→ $-0.5 \leq x_2 \leq 0.5$

* Scale *

Making Sure gradient descent is working Correctly.

1. Plot $J(\theta)$ as a Function of the number of iterations.
2. It works Correctly When $J(\theta)$ decreases after every iteration.

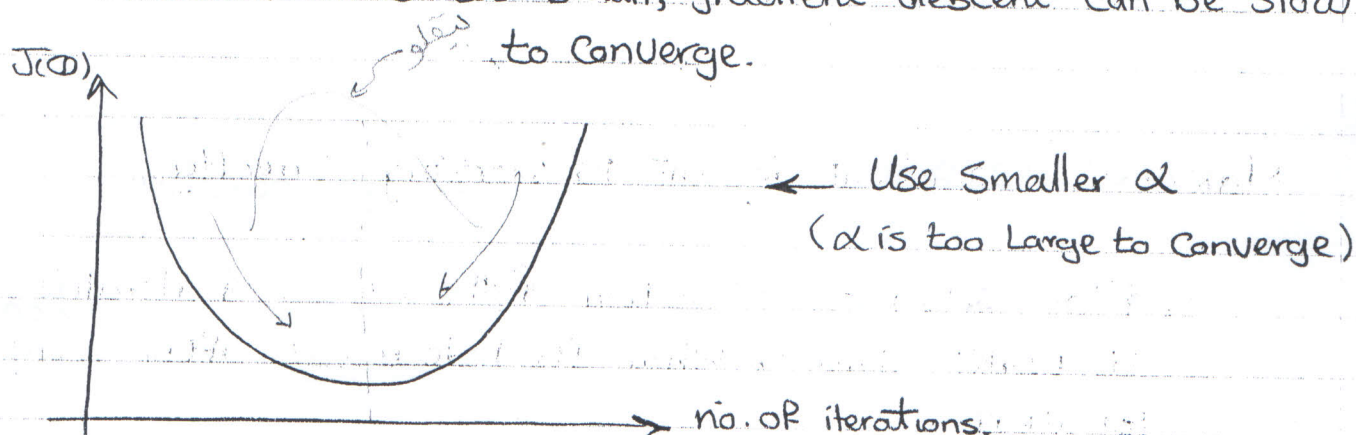




Here, Gradient descent not working.

3. For sufficiently small α , $J(w)$ should decrease on every iteration.

. But if α is too small, gradient descent can be slow to converge.



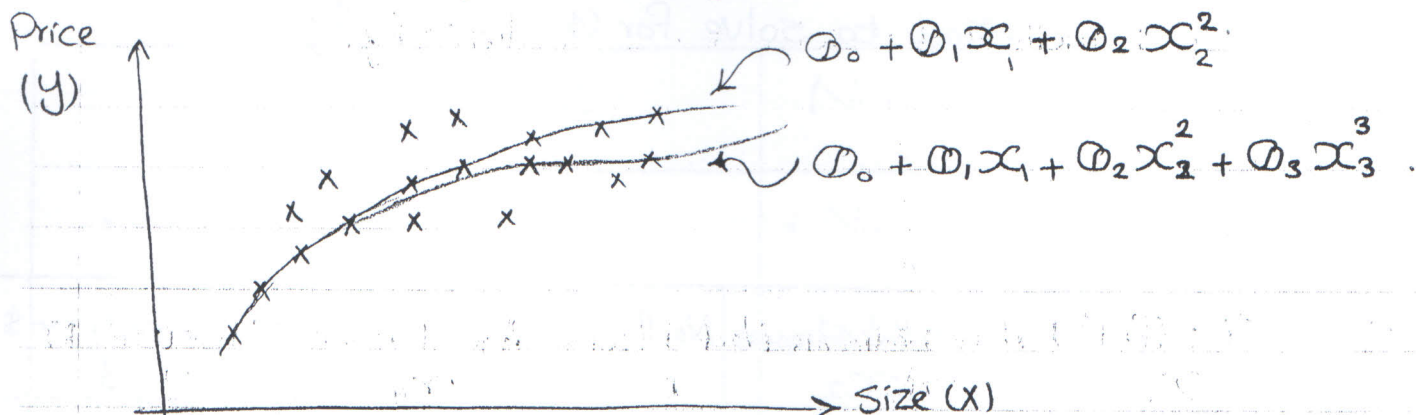
Summary:

- IF α is too small : Slow Convergence.
- IF α is too Large : $J(w)$ may not decrease on every iteration ; may not Converge.

To Choose α , Try

---, 0.001, ---, 0.01, ---, 0.1, ---, 1, ---

Features and Polynomial Regression.



$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$h_{\theta}(x) = \theta_0 + \theta_1 (\text{Size}) + \theta_2 (\text{Size})^2 + \theta_3 (\text{Size})^3 \quad (\text{Size})^{\text{Size}} \text{ and } \text{Feature}$$

$$x_1 = \text{Size}, \quad x_2 = (\text{Size})^2, \quad x_3 = (\text{Size})^3.$$

• Normal Equation

: method to solve for θ analytically.

x_0	Size (feet ²) x_1	No. of bedrooms x_2	No. of Floors x_3	Age of home x_4	Price (1000\$) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178
1	3000	4	1	38	5

$$\theta = (X^T X)^{-1} X^T y$$

المصفوفة θ (المعلمة)

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \\ 1 & 300 & 4 & 1 & 38 \end{bmatrix} \quad , \quad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \\ 540 \end{bmatrix}$$

المصفوفة X (المصفوفة المربعة)

→ m training examples (عدد الصفوف)

→ n Features (عدد الأعمدة)

Gradient Descent	Normal Equation.
<ul style="list-style-type: none"> • Need to Choose α • Needs many iterations. • <u>Works well even when</u> <u>n is large.</u> 	<ul style="list-style-type: none"> • No need to Choose α. • Don't need to iterate. • Need to Compute $(X^T X)^{-1}$ • <u>Slow IF n is very large.</u> لو m (training set) كبير